

1           CLAIMS

2

3       **1.** A system comprising:

4           a content host configured to receive multimedia contents for delivery to a  
5 content requester over a network, and to process the multimedia contents to  
6 provide an abstract content model that represents the multimedia contents; and

7           a decision engine communicatively linked with the content host and  
8 configured to receive the abstract content model and process the abstract content  
9 model to find an optimized sub-graph, and to select a delivery plan for the  
10 multimedia contents based at least in part on the optimized sub-graph.

11  
12       **2.** The system of claim 1, wherein the content host is configured to  
13 process the multimedia contents to provide the abstract content model responsive  
14 to receiving a request for content from the content requester.

15  
16       **3.** The system of claim 1, wherein the abstract content model is  
17 configured to hide physical details of the multimedia contents.

18  
19       **4.** The system of claim 1 further comprising a resource model module  
20 configured to model network characteristics and to be used by the decision engine  
21 to select a delivery plan.

1       5. The system of claim 1 further comprising a preference model module  
2 configured to model client capabilities and to be used by the decision engine to  
3 select a delivery plan.

4

5       6. The system of claim 1 further comprising:  
6           a resource model module configured to model network characteristics and  
7 to be used by the decision engine to select a delivery plan; and  
8           a preference model module configured to model client capabilities and to  
9 be used by the decision engine to select a delivery plan.

10

11       7. The system of claim 1, wherein the content host defines a set of  
12 application programming interfaces for retrieving extended properties of the  
13 abstract content model.

14

15       8. The system of claim 1, wherein the decision engine is content  
16 independent.

17

18       9. The system of claim 1 further comprising a content mapper module  
19 that is configured to process the abstract content model to provide multimedia  
20 content for delivery in accordance with the delivery plan.

21

22       10. The system of claim 1, wherein the decision engine is configured to  
23 find the optimized sub-graph in a manner that maximizes one or more quality of  
24 service (QoS) values.

1       **11.**   The system of claim 1, wherein the decision engine is configured to  
2 find the optimized sub-graph in a manner that maximizes one or more quality of  
3 service (QoS) values at least some of which being associated with resource  
4 constraints.

5  
6       **12.**   The system of claim 1, wherein the decision engine is configured to  
7 find the optimized sub-graph in a manner that maximizes one or more quality of  
8 service (QoS) values at least some of which being associated with client  
9 preference factors.

10  
11      **13.**   The system of claim 1, wherein the decision engine is configured to  
12 find the optimized sub-graph in a manner that maximizes one or more quality of  
13 service (QoS) values at least some of which being associated with resource  
14 constraints and others of which being associated with client preference factors.

15  
16      **14.**   The system of claim 1 further comprising a cache model configured  
17 to save at least portions of one or more selected delivery plans for reuse in other  
18 content request situations.

19  
20      **15.**   One or more computer-readable media embodying the system of  
21 claim 1.

22  
23      **16.**   One or more server computers embodying the system of claim 1.

1       **17.** A method comprising:

2           receiving a content request from a content requester;

3           retrieving the requested content from a content source;

4           processing the retrieved content to provide an abstract content model;

5           processing the abstract content model to select an optimal delivery plan the  
use of which will permit requested content to be delivered to the content requester;

6

7       and

8           processing the abstract content model to provide deliverable content in  
9           accordance with the selected delivery plan.

10

11       **18.** The method of claim 17, wherein said processing of the retrieved  
12           content to provide the abstract content model comprises providing at least one  
13           directional graph.

14

15       **19.** The method of claim 18, wherein the directional graph features a  
16           top-down hierarchical structure.

17

18       **20.** The method of claim 18, wherein the directional graph comprises  
19           nodes that represent components of the content and edges that represent  
20           relationships between the nodes.

21

22       **21.** The method of claim 20, wherein the nodes are configured to have a  
23           node status that defines dynamic statuses of nodes during content delivery.

1       **22.** The method of claim 21, wherein at least one status comprises an  
2 inactive status where the node is not yet a deliverable object.

3  
4       **23.** The method of claim 21, wherein at least one status comprises an  
5 activable status wherein an active condition of the node is satisfied but the node is  
6 not yet included in a delivery plan.

7  
8       **24.** The method of claim 21, wherein at least one status comprises an  
9 activated status wherein the node has been chosen in a delivery plan.

10  
11      **25.** The method of claim 21, wherein at least one status comprises a  
12 delivered status wherein the node has been delivered successfully to a content  
13 receiver.

14  
15      **26.** The method of claim 21, wherein at least one status comprises a  
16 skipped status wherein the node is not delivered and will not be included in the  
17 delivery plan.

18  
19      **27.** The method of claim 20, wherein there are multiple different types  
20 of edges.

21  
22      **28.** The method of claim 27, wherein the multiple different types of  
23 edges comprise a dependency edge type that defines a logical dependency between  
24 nodes.

1           **29.** The method of claim 27, wherein the multiple different types of  
2 edges comprise a route edge type that defines an ordered or hierarchical  
3 dependency between nodes.

4

5           **30.** The method of claim 27, wherein the multiple different types of  
6 edges comprise a mixed edge type that defines a logical dependency between  
7 nodes and an ordered or hierarchical dependency between nodes.

8

9           **31.** The method of claim 27, wherein the multiple different types of  
10 edges comprise one or more of the following:

11                 a dependency edge type that defines a logical dependency between nodes,  
12                 a route edge type that defines an ordered or hierarchical dependency  
13                 between nodes, and  
14                 a mixed edge type that defines a logical dependency between nodes and an  
15                 ordered or hierarchical dependency between nodes.

16

17           **32.** The method of claim 20, wherein the directional graph comprises a  
18 directional graph  $G=(N, E)$  that satisfies the following condition, where  $N$  and  $E$   
19 stand for “node” and “edge” sets of G:

20                 • The node set  $N$  can be divided into several subsets as  $N_i$ ,

21                  $i=1..m$  where  $N = \bigcup_{i=1}^m N_i$     and     $N_i \cap N_j = \emptyset$ ,

22                  $\forall i, j = 1..m, i \neq j$ .

23                 •  $\forall a = (n_s, n_e) \in E$ , there exists a pair of  $1 \leq i < j \leq m$  such that  $n_s \in N_i$ ,  
24                     and  $n_e \in N_j$ .

25                 •  $\forall a = (n_s, n_e) \in E, s \neq e$ .

1       **33.** A server computer programmed to implement the method of claim  
2       17.

3       **34.** A method comprising:

4              processing content that can be the subject of a content request from a  
5              content requester to provide an abstract content model;

6              processing the abstract content model to select at least one delivery plan the  
7              use of which will permit requested content to be delivered to a content requester;  
8              and

9              processing the abstract content model to provide deliverable content in  
10             accordance with the selected delivery plan.

11       **35.** The method of claim 34, wherein said processing of the retrieved  
12             content to provide the abstract content model comprises providing at least one  
13             directional graph.

14       **36.** The method of claim 35, wherein the directional graph features a  
15             top-down hierarchical structure.

16       **37.** The method of claim 35, wherein the directional graph comprises  
17             nodes that represent components of the content and edges that represent  
18             relationships between the nodes.

19       **38.** A method comprising:

20              receiving a content request from a content requester;

1 retrieving the requested content from a content source;

2 processing the retrieved content to provide an abstract content model  
3 comprising a directional graph featuring a top-down hierarchical structure having  
4 nodes that represent components of the content and edges that represent  
5 relationships between the nodes, the nodes being configured to have a node status  
6 that defines dynamic statuses of nodes during content delivery, the node statuses  
7 being selected from a group of statuses comprising: (1) inactive status where the  
8 node is not yet a deliverable object, (2) activable status wherein an active  
9 condition of the node is satisfied but the node is not yet included in a delivery  
10 plan, (3) activated status wherein the node has been chosen in a delivery plan, (4)  
11 delivered status wherein the node has been delivered successfully to a content  
12 receiver, and (5) skipped status wherein the node is not delivered and will not be  
13 included in the delivery plan; and wherein there are multiple different types of  
14 edges selected from a group of types comprising: (1) a dependency edge type that  
15 defines a logical dependency between nodes, (2) a route edge type that defines an  
16 ordered or hierarchical dependency between nodes, and (3) a mixed edge type that  
17 defines a logical dependency between nodes and an ordered or hierarchical  
18 dependency between nodes;

19 processing the abstract content model to select an optimal delivery plan the  
20 use of which will permit requested content to be delivered to the content requester;  
21 and

22 processing the abstract content model to provide deliverable content in  
23 accordance with the selected delivery plan.

1           **39.**   The method of claim 38, wherein the abstract content model  
2 comprises an ignition edge that is defined as a dependency edge from a node that  
3 is activated, delivered or skipped.

4  
5           **40.**   The method of claim 39, wherein nodes have active conditions that  
6 define how a node becomes activable.

7  
8           **41.**   The method of claim 39, wherein nodes have active conditions that  
9 define how a node becomes activable, one of the conditions comprising an  
10 automatic condition wherein the node is automatically activable.

11  
12          **42.**   The method of claim 39, wherein nodes have active conditions that  
13 define how a node becomes activable, one of the conditions comprising an OR  
14 condition wherein the node is activable if at least one of its input edges is an  
15 ignition edge.

16  
17          **43.**   The method of claim 39, wherein nodes have active conditions that  
18 define how a node becomes activable, one of the conditions comprising an AND  
19 condition wherein the node is activable only if all its input edges are ignition  
20 edges.

21  
22          **44.**   The method of claim 39, wherein nodes can have input conditions  
23 that define when an activable node can be activated.

1           **45.**   The method of claim 39, wherein nodes can have output behaviors  
2        that define how nodes on ends of outgoing route edges can be branched in  
3        accordance with a branching operation, where a branched node constitutes a node  
4        that has been changed from an activable node to an activated node.

5  
6           **46.**   The method of claim 45, wherein there are multiple different  
7        branching operations.

8  
9           **47.**   The method of claim 46, wherein one of the multiple different  
10      branching operations comprises a complementary branching operation wherein all  
11      activable nodes can be branched.

12  
13          **48.**   The method of claim 46, wherein one of the multiple different  
14      branching operations comprises an exclusive branching operation wherein only  
15      one node from all activable nodes can be branched.

16  
17          **49.**   The method of claim 46, wherein one of the multiple different  
18      branching operations comprises a tight branching operation wherein all activable  
19      nodes must be branched.

20  
21          **50.**   The method of claim 38, wherein nodes can have a value associated  
22      with a Quality of Service (QoS) factor which defines an increment to content  
23      quality when this node has been delivered.

1       **51.**   The method of claim 38, wherein nodes can have a resource factor  
2 that defines an amount of resources needed to deliver the node.

3  
4       **52.**   A data structure comprising:

5              an abstract content model that represents content that can be requested from  
6 a content requester, the abstract model comprising a directional graph that features  
7 a top-down hierarchical structure having nodes that represent components of the  
8 content and edges that represent relationships between the nodes, the nodes being  
9 configured to have a node status that defines dynamic statuses of the nodes during  
10 content delivery;

11             the abstract content model being configured for processing so that an  
12 optimal delivery plan for delivering the content can be selected.

13  
14       **53.**   The data structure of claim 52, wherein at least one status comprises  
15 an inactive status where the node is not yet a deliverable object.

16  
17       **54.**   The data structure of claim 52, wherein at least one status comprises  
18 an activable status wherein an active condition of the node is satisfied but the node  
19 is not yet included in a delivery plan.

20  
21       **55.**   The data structure of claim 52, wherein at least one status comprises  
22 an activated status wherein the node has been chosen in a delivery plan.

1       **56.**     The data structure of claim 52, wherein at least one status comprises  
2 a delivered status wherein the node has been delivered successfully to a content  
3 receiver.

4

5       **57.**     The data structure of claim 52, wherein at least one status comprises  
6 a skipped status wherein the node is not delivered and will not be included in the  
7 delivery plan.

8

9       **58.**     The data structure of claim 52, wherein there are multiple different  
10 types of edges.

11

12       **59.**     The data structure of claim 58, wherein the multiple different types  
13 of edges comprise a dependency edge type that defines a logical dependency  
14 between nodes.

15

16       **60.**     The data structure of claim 58, wherein the multiple different types  
17 of edges comprise a route edge type that defines an ordered or hierarchical  
18 dependency between nodes.

19

20       **61.**     The data structure of claim 58, wherein the multiple different types  
21 of edges comprise a mixed edge type that defines a logical dependency between  
22 nodes and an ordered or hierarchical dependency between nodes.

1       **62.**   The data structure of claim 58, wherein the multiple different types  
2 of edges comprise one or more of the following:

3              a dependency edge type that defines a logical dependency between nodes,

4              a route edge type that defines an ordered or hierarchical dependency

5 between nodes, and

6              a mixed edge type that defines a logical dependency between nodes and an  
7 ordered or hierarchical dependency between nodes.